International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:1, 2012

A New Protocol for Concealed Data Aggregation in Wireless Sensor Networks

M. Abbasi Dezfouli, S. Mazraeh, M. H. Yektaie

***Abstract*—**Wireless sensor networks (WSN) consists of many sensor nodes that are placed on unattended environments such as military sites in order to collect important information. Implementing a secure protocol that can prevent forwarding forged data and modifying content of aggregated data and has low delay and overhead of communication, computing and storage is very important. This paper presents a new protocol for concealed data aggregation (CDA). In this protocol, the network is divided to virtual cells, nodes within each cell produce a shared key to send and receive of concealed data with each other. Considering to data aggregation in each cell is locally and implementing a secure authentication mechanism, data aggregation delay is very low and producing false data in the network by malicious nodes is not possible. To evaluate the performance of our proposed protocol, we have presented computational models that show the performance and low overhead in our protocol.

***Keywords*—**Wireless Sensor Networks, Security, Concealed Data Aggregation.

I. Introduction

A Wireless sensor network is composed of large number of sensor nodes that have strictly limited computation and communication abilities and power resources [1]. In the near future, wireless sensor networks are envisioned to be employed widely in many applications including critical area surveillance, home and office automation, habitat monitoring, health monitoring, and military tracking. Therefore, security is an essential issue in wireless sensor networks and widespread deployment of these networks could be curtailed without adequate security [2], [3]. However, compared to conventional computer networks, implementing security is not easy in wireless sensor networks due to limited processing power, storage, bandwidth, and energy of sensor nodes. In addition to security, limited battery power and bandwidth of sensor nodes make it a challenging task to provide efficient solutions to data gathering problem. Therefore, in order to reduce the power and bandwidth consumption of wireless sensor networks, several mechanisms are proposed such as data aggregation [4]. Data aggregation protocols aim to combine and summarize data packets of several sensor nodes so that overall in network communication bandwidth and energy consumptions are reduced.

Since data aggregation and security are essential for wireless sensor networks, providing secure data aggregation has been an attractive problem for researchers [5], [6], [7], [8], [9], [10], [11].

M. Abbasi Dezfouli is Associate Professor and Head of Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Khouzestan, Iran (m.abbasi@khouzestan.srbiau.ac.ir)

S. Mazraeh is postgraduate student in Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Khouzestan, Iran. (s.mazraeh@siau.ac.ir)

M. H. Yektaie is Associate Professor at Department of Computer Engineering, Abadan, Islamic Azad University, Khouzestan, Iran. (mh.yektaie@gmail.com)

In many of the existing secure data aggregation protocols, data aggregators must decrypt every message they receive, aggregate the messages according to the corresponding aggregation function, and encrypt the aggregation result before forwarding it. Therefore, while these data aggregation protocols improve the bandwidth and energy utilization in the network, they negatively affect other performance metrics such as delay and security. To support secure data aggregation without causing delay, a set of data aggregation protocols is proposed. These protocols use privacy homomorphic encryption to allow data aggregation without requiring decryption of the data [12], [13], [14]. Protocols in [12] and [13] utilize symmetric and asymmetric privacy homomorphic encryption to allow aggregation of encrypted data, respectively. However, in [12], sensor data must be encrypted with a single key to perform concealed data aggregation. Therefore, in order to hierarchically aggregate data of the whole network, sensor nodes in the network must share a common key and use it for encryption. Using a single symmetric key in the network is not secure as an adversary can fake the aggregated results through compromising only a sensor node. In addition, symmetric key based privacy homomorphism is shown to be insecure for chosen plaintext attacks for some specific parameter settings [15]. The scheme proposed in [13] relies on asymmetric key based privacy homomorphism but it also requires a single public key to allow hierarchical data aggregation. The scheme proposed in [14] allows using different encryption keys in aggregated data. Authors employ an extension of the one-time pad encryption technique using additive operations modulo n. However, several practical issues are not addressed in this paper such as requirement of a strong synchronization mechanism.

In this paper, we propose Hierarchical Concealed Data Aggregation (HCDA) protocol which allows concealed aggregation of data that are encrypted with different keys. HCDA protocol virtually partitions the network into several regions and employs a different public key in each region. Due to the privacy homomorphic encryption scheme [20] of HCDA, the data collected in a region can be encrypted using the public key of the region and the encrypted data of several regions can be hierarchically aggregated into a single piece of data without violating data confidentiality. Moreover, during the decryption of aggregated data, the base station is able to determine the origin of the data based on the encryption key. This is particularly useful when the base station needs data from a certain region of the network. In order to use multiple keys in the network area, HCDA protocol employs a group based network deployment scheme where sensor nodes in a group use the same public key. In addition, as HCDA protocol is based on elliptic curve cryptography, it is not affected by node compromise attacks whereas symmetric key based concealed data aggregation protocols [12] are significantly affected from

these attacks. Our theoretical analysis shows that HCDA is feasible for resource constrained sensor nodes.

Our contribution in this work is that we provide a concealed data aggregation technique that allows hierarchical aggregation of data encrypted with different keys. Note that to the best of our knowledge this property cannot be efficiently achieved by any other existing concealed data aggregation scheme.

The rest of the paper is organized as follows. In Section II, the state-of-the-art in secure data aggregation is presented. Section III explains the system model and preliminaries along with HCDA's network deployment scenario. HCDA protocol is given in Section IV. Concluding remarks are made in Section V.

II. Related Work

In wireless sensor network domain, secure data aggregation problem is studied extensively [5], [6], [7], [8], [9], [10], [11]. In [5], the security mechanism detects node misbehaviors such as dropping or forging messages and transmitting false data. In [6], random sampling mechanisms and interactive proofs are used to check the correctness of the aggregated data at the base station. In [8], witness nodes of data aggregators also aggregate data and compute MACs to help verify the correctness of the aggregators' data at base station. Because the data validation is performed at base station, the transmission of false data and MACs up to base station affects adversely the utilization of sensor network resources. In [9], sensor nodes use the cryptographic algorithms only when a cheating activity is detected. Topological constraints are introduced to build a secure aggregation tree (SAT) that facilitates the monitoring of data aggregators. In [10], a Secure Hop-by-hop Data Aggregation Protocol (SDAP) is proposed. The authors of SDAP are motivated by the fact that, compared to low level sensor nodes, more trust is placed on the high-level nodes (i.e., nodes closer to the root) during a normal hop-by-hop aggregation process in a tree topology. In [11], the authors propose a protocol that makes use of a web of trust to overcome the shortcomings of cryptography based secure data aggregation solutions.

Privacy homomorphism is introduced by Rivest et al. [16]. For example, Rivest's asymmetric key algorithm RSA is multiplicatively homomorphic. Due to their high computational overhead, such asymmetric key homomorphic encryption algorithms are not feasible for sensor nodes [17]. The privacy homomorphic encryption algorithm introduced by Domingo- Ferrer [18] is symmetric key based. The concealed data aggregation algorithm that is proposed in [12] employs Domingo- Ferrer's privacy homomorphic encryption algorithm. However, in order to hierarchically aggregate the data of the all network, the proposed scheme must uses a secret key known by all sensor nodes which leads to the following attack. If a sensor node is compromised, it can decrypt data of any sensor node which is encrypted by the secret key. Hence, in this paper, we use a privacy homomorphic function that is based on elliptic curve cryptography (ECC). Compared to RSA, ECC provides the same security level with shorter key size and ciphertexts. It is shown that 160-bit ECC key provides the same security as 1024-bit RSA key provides [19]. Since communication overhead of wireless sensor networks depends on the size of data packets, ECC based privacy homomorphic encryption schemes are more preferable.

III. Suggested Protocol

In our suggested protocol in this paper, it is assumed that all sensor nodes, with the exception of central node, have limited in storage, Computation and communication and all of them are vulnerable to the attack of adversaries. Only the central node is invulnerable to the attacks of adversaries. After network deployment, every node can be informed of its position by global positioning system (GPS). Since the network is unknown to the adversary, it is logical to assume that the network and sensor nodes would not be attacked in the network setup step and before the start of network operation [6]. Sensor nodes are connected to each other in a multihop manner. Central node stands in (0, 0) position of the network.

A) Assumptions and abbreviations

In Table I, the abbreviations which are used in our protocol have been presented.

Table I
Abbreviations used in the suggested protocol

Symbol	Description
L_m	Primary key for keeping confidentiality
K_s	Slave key for authentication
K_{auth}	Unique authentication key
K_{enc}	Unique encryption key
$K_{auth\text{-}cell}$	Cell-internal authentication key
$K_{enc\text{-}cell}$	Cell-internal secret key
$F_x(Y)$	Semi-random function for key generation
D_{agg}	Aggregated data
Enc[]	Encryption
Dec[]	Decryption
MAC[]	Making message authentication code

B) System Model and Preliminaries

Due to concealed data aggregation protocols, routing and secret keys distribution are closely related to each other, the suggested protocol in this paper includes various steps in which there questions have been taken into consideration. In the following pages, each part of our protocol will be presented and explained separately.

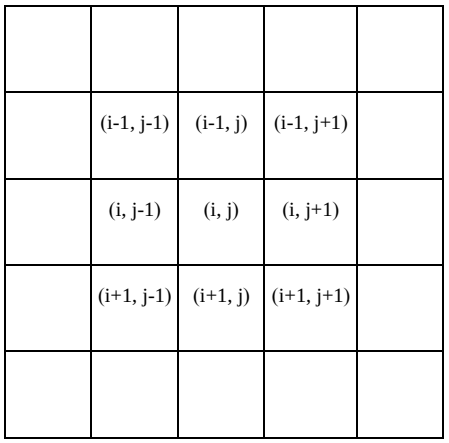

Fig. 1 Numbering cells in virtual mesh

In network setup step, central node before the distribution of sensor node in the network, puts the primary keys (K_m, K_s) in the memory of all nodes. In this way, using these keys, sensor nodes can connect safely to each other and to central node. Also, the operators of aggregation – like sum, average, and aggregation function – distribute the data in the memories of nodes. The network is considered to be like a virtual mesh which includes virtual cells. In each cell, there are some sensor nodes. Each cell, as a cluster in the network, is used for data aggregation. Shape and size of these cells permits us to consider the nodes of adjacent cells to be as neighbors. The distance between two nodes in two neighboring cells should be equal to R (range of node's transference). This is done by selecting an appropriate length for cell-sides.

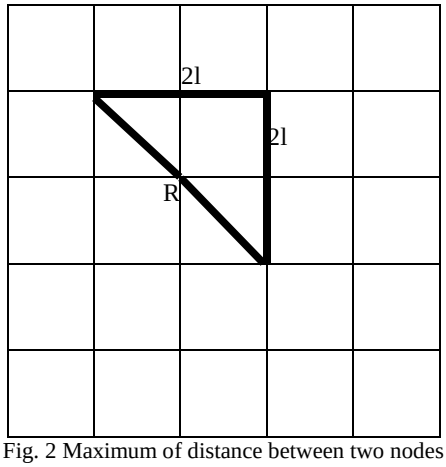

Fig. 2 Maximum of distance between two nodes

Fig. 1shows the manner of neighboring cells numbering and Fig. 2 shows the maximum of distance between two nodes in two neighboring cells. Considering the range of node transference, we can calculate the length of cell-side by (1). *l* is the length of cell-side.

$$R^2 = (2l)^2 + (2l)^2 \Rightarrow l = \frac{R}{\sqrt{8}} \quad (1)$$

After the deployment of nodes in the network, considering this logical assumption that the network has not been operated and data have not been sent to central node in the network setup step and authentication, we can suppose that the advisory would not be able to attack sensor node and to compromise. Each node determines a certain time period for authentication and network setup which is called $t_{deadline}$. This time period should be equal to or less than the time for network setup.

The length of this time has been distributed by central node in this period, sensor nodes can create routing table. At the onset, each ID-node emits its energy and physical positions in its neighbor hood. After the sending of this message by all nodes, they are informed in which cell they have been put and which nodes are in the same cell. This is done by considering cell-sides and the assumption of nodes. The node which has the maximum amount of energy is selected as the first header of the cell. Only the first headers of the cells participate in the creation of routing tables. This causes the reduction of energy consumption in the network. In the next step, the central node sends a package across the network. This package is called start_p. this massage includes two cell-number fields and the number of steps toward the central node. Receiving this message, the first headers of cells store the data in their routing table. Putting the cell-number in start_p package which has been sent from central node, the first headers update the package and sent it across the network. Receiving the start_p package, the network nodes compare the number of steps to central node with the number recorded from routing table, the number in the routing table and it will replace the number in the routing table and it the package are updated and spread across the network. If the number of steps to the central node is equal to the recorded number from routing table, the data included in the package will be registered in the routing table. If these two conditions do not occur, the package will be discarded. By using (2) the node *i*, which is in a cell with *n* node, can compare its energy with the energy of other nodes in the same cell and candidates itself as header of the cell.

$$\forall i \ \ i \le n \qquad P_i = \frac{Enlev_i}{\sum_{i=1}^{n} Enlev_i} \quad (2)$$

In this equation, E_{nlev} is the level of remained energy at that time and *P* is the probability of selecting node as header. Using afore said equation, two nodes which have the maximum amount of energy in the cell will be selected as the second and third headers. Sending a message to the nodes of their cells, the headers will announce their headness and the hierarchy of this headness. Then, each node begins to make the necessary keys for safe connections. $K_{enc}=F_{km}$ (ID_i) is a key for unique codification that each node uses to contact the central node. ID_i is the same as the identification of node *i*.

$K_{auth}=F_{ks}(ID_i)$ is a key for producing the message of unique authentication during the connection of node *i* with Central node.

$K_{enc-cell}=F_{km}$ (ID_{cell}) is the key for cell internal codification which is used in codifying the connections between the nodes in each cell.

$K_{auth-cell}=F_{ks}(ID_{cell})$ is the key for producing messages of cell-internal authentication. After the creation of these keys by each node and before the finishing of $t_{deadline}$, each node can delete K_m and K_s from its memory. If it is not done, the data of node-keys will be in danger because the adversary can gain access to the nodes during the sending of data. In this way, using the semi-random function, the adversary can produce all keys of the nodes. After network setup, using $K_{enc-cell}$, the cell-internal nodes codify their data. Also, using $K_{auth-cell}$, the cell-internal nodes codify their data. Also, using $K_{auth-cell}$, they produce the message of authentication and send both messages in the form of a package to the first, second, and third headers.

In the steps of data aggregation and sending to central node, after decryption and authentication of sender of each package and by using $K_{auth-cell}$, the headers of each cell will be determined. (3), (4) show the production of aggregated message after codification and also the production of authentication message in the i_{th} header of each cell.

$$Enc[K_{enc}(D_{agg}, nonce)] \quad (3)$$

$MAC[K_{auth}(D_{agg})]$ (4)

In (3) *nonce* is a random value which is sent to the central node by each header. By doing this, they prevent the occurrence of replication attack. In this attack, the attackers reproduce the same data and send them to destination with a false address. By doing this, they cause disorder in the traffic of the network and mislead the receiver. Since the attacker is not aware of the value of *nonce-less* message, it is invalid and discarded. The first header of each cell sends a package which includes a message shown in (3) and (4). $Enc[K_{enc}(D_{agg} \| nonce \| MAC[K_{auth}(D_{agg})])]$ is the form of package which is sent to the central node. Since routing in the network occurs in a cross-cell manner, the package of data is sent to the first header of neighboring cell. The route of this package is traced by the first header in the shortest course to central station. Since the central node has the ability to produce the related keys, it decodes the data by using K_{enc}. Also, by using the related K_{auth}, it sends the data to the node. Since routing in the network occurs in a cross-cell manner, the package of data is sent to the first header of neighboring cell. The route of this package is traced by the first header in the shortest course to central station. since the central node has the ability to produce the related keys, it decodes the data by using K_{enc}. Also, by using the related K_{auth}, it sends the data to the node. By producing $MAC[K_{auth}(D_{egg})]$ and making a comparison with MAC message which is sent by the sender, it can realize the identity of the sender.

The packages sent by second and third headers are received by second and third headers respectively in neighboring cell. This is done by using the data in the routing table. Then, these packages are directed to the central node.

After receiving packages sent by the headers of each cell, the central node should be assured that the data are not false and forgery; because the adversary might have gained access to the headers, especially the first header. So, it compares the MAC message received from all three headers. Then, the following situation occurs:

- All of the three MAC messages are confirmed: In this situation, the data are confirmed and all headers are considered to have done their job properly.

Two out of three messages have integer value and the other non-integer: in this situation, if the MAC message sent by the first header is confirmed, the sent package of data will be confirmed. Then, the central node sends a message to first header and reports the malfunction of the headers which have not done their work properly. These headers will be deleted from the list and first header selects a node from the cell randomly. The energy of the selected node should be more than threshold energy which is an adaptable parameter. This node replaces the node which had not done its work properly. But, when the first header does not do properly, the aggregated data would not be confirmed by central node and would be discarded. By sending a message to the second header, the central node selects the second header as the first header and asks it to send the package which includes $Enc[K_{enc}(D_{agg} \| nonce \| MAC[K_{auth}(D_{agg})])]$. In this condition, the central node receives the right data. The new first header randomly selects a node in the cell as the second header. The energy of new second header should be more than threshold energy. The new first header sends a message to all nodes of its cell and announces its headness and the value of header's hierarchy. If new node is added in the step of sending data across the network, central node will pre-distribute related K_{enc} and K_{auth} in its memory. Also, it stores the data related to cells and threshold energy in its memory. After network deployment and by using the data about its physical position, the node identifies its cell. Then it sends a message to the central node and asks for a cell-internal key. This message includes the information about its cellular position in the form of a encrypted message which has been authenticated. After receiving this message from related node, the central node encrypts $K_{ens\text{-}cell}$ and $K_{auth\text{-}cell}$. This encryption is done by using K_{enc} and K_{auth}. Then, the message is sent to target node.

In this part, we want to discuss the form of selecting a new header in each cell when one of the headers, for some reasons, has prevented it's energy to reach a level lower than $t_{deadline}$ to prevent the increase in the number of messages, which include information about replacing headers, we are going to present a new method. In this method, when energy reaches to a level lower than threshold limit, the deader randomly sends a message to one of the nodes in the cell with this condition that the node should not be header of the cell. In this period, if they remained energy of the selected node is more than threshold limit and it considers itself as an eligible header, it will send a message to its cell-internal nodes and will announce itself as headness. Receiving this message, the previous header node is informed about the selection of new header. When new header disqualifies itself as a header, it sends a message to previous header and announces its disqualification. Then, the previous header sends a new message to another node and starts the process of selecting a new header.

IV. Analytical Evaluation and Simulation Results

In this part, in addition to analytical evaluation, we evaluate the simulation results based on its energy consumption overhead.

A) Evaluating protocol based on security requirements

In table II, the results of comparison between the suggested protocol in this paper with protocols of concealed data aggregation have been presented.

TABLE II
COMPARING SUGGESTED PROTOCOL WITH OTHER SUGGESTED PROTOCOLS IN DATA AGGREGATION BASED ON SECURITY REQUIREMENTS

Protocols	Confidentiality	Data integrity and freshness	Source authentication	availability
SRDA(4)	✓	✓	✓	
SDAP(8)	✓	✓	✓	
SELDA(9)		✓	✓	✓
Secure DAV(2)	✓	✓	✓	
Du et al. [6]	✓	✓	✓	
Suggested protocol	✓	✓	✓	✓

As mentioned in table II, the suggested protocol satisfies all security requirements. The exchanged data among nodes in each cell are encrypted, this meets the confidentiality needs.

The nonce values included in sent packages guarantees the integrity and data freshness of each node of aggregated data. In this way, it prevents replication attack. By using this protocol, attacks such as Sybil attacks are prevented. This is done by message authentication codes which are included in sent packages. Considering the local selection of aggregated nodes in each cell, the availability needs will be met.

B) Analytical evaluation based on attack on aggregation nodes

The invulnerability of this protocol against the aggregating nodes is of great importance.

$$P(T) = \left(\frac{\binom{a_i}{3}}{\binom{n_i}{3}} + \frac{\binom{a_i}{2} \times \binom{c_i}{1}}{\binom{n_i}{3}} \right)^m , n_i = \frac{N}{m}; \; i = 0,1,\ldots, \; m \qquad (5)$$

P(T) in (5) shows the probability of compromise of, at most, One of the headers in the network. This is equal to the probability of occurrence of a situation in which among the headers of each cell, at most, one header is compromised by adversary. In this equation, n_i is the number of nodes in cell *i*. a_i is the number of permissible nodes in cell *i* and C_i is the number of compromised nodes in cell *i*. *m* is the number of all cells in the network and *N* is the number of nodes in the network, including permissible and compromised nodes. Consider the condition in which one of the header nodes in each cell is compromised. If the probability of this conditions are added each other, *P(T)* will be calculated. This is the probability of the worst situation; because in this condition, in all cells simultaneously, at most, one node has been compromised. Fig. 3 shows the results gained from (5) for a 10-cell network in which the sensor nodes have been distributed uniformly in the cell.

C) The evaluating operation based on simulation results

In this part, we are going to discuss the operation of protocol on the basis of simulation results. In order to simulate, we have used J-SIM [10] software. In the performed scenarios, 450 nodes have been put randomly in an area of 400m×400m. The length of each cell-side is 40m. The results of this simulation have been compared with [6] and [9] protocol. 660mw and 395mw are the amounts of energy which has been consumed to send and receive data respectively [11]. The compromised nodes, randomly, are distributed in the network from the beginning of simulation. The evaluation criteria are correction value of aggregated data error in the central node and the mean of consumed energy in the suggested protocol.

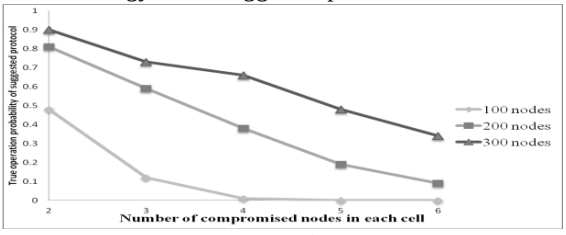

Fig. 3 True operation probability of suggested protocol in attack situation and compromised node existence in the network

D) The evaluating operation based on

In Fig. 4, for receiving every integer aggregated data, the amount of consumed energy has been presented. This is calculated by using the ratio of the mean of consumed energy to the mean of received integer data in the central node.

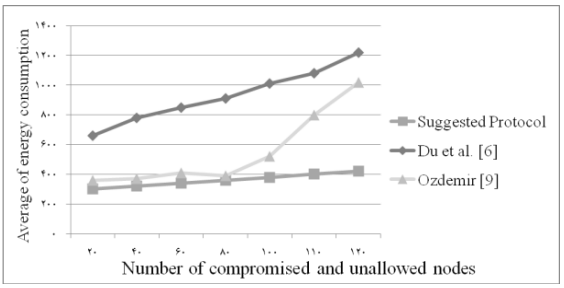

Fig. 4 Amount of energy consumption for data aggregated protocols

As you can see, since data aggregation and sending to central node is performed by headers of cells, the amount of consumed energy in our suggested protocol is much less than the amount of consumed energy in source protocol. When unallowed nodes are increased in the network, this reduction in energy consumption is more remarkable.

V. Conclusion

In this paper, we have presented our ongoing work on hierarchical concealed data aggregation. Considering to data aggregation in each cell is locally and implementing a secure authentication mechanism, data aggregation delay is very low and producing false data in the network by malicious nodes is not possible. Currently, we are working on adding an integrity check mechanism to proposed scheme and implementing the proposed scheme to evaluate its security and performance.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, 40(8), pp. 102- 114, Aug. 2002.

[2] H. Chan and A. Perrig, "Security and Privacy in Sensor Networks", IEEE Comp. Mag., Oct. 2003, pp. 10305.

[3] E. Shi and A. Perrig, "Designing Secure Sensor Networks", Wireless Commun. Mag., vol. 11, no. 6, Dec. 2004 pp. 3843.

[4] K. Akkaya, M. Demirbas, and R. S. Aygun, "The Impact of Data Aggregation on the Performance of Wireless Sensor Networks", Wiley Wireless Communications and Mobile Computing (WCMC) Journal, Vol. 8 pp. 171-193, 2008.

[5] L. Hu and D. Evans, "Secure aggregation for wireless networks", Proc. of Workshop on Security and Assurance in Ad hoc Networks, Jan 28, Orlando, FL, 2003.

[6] B. Przydatek, D. Song, and A. Perrig, "SIA : Secure information aggregation in sensor networks", Proc. of SenSys'03, pp. 255-265, 2003.

[7] H. C¸ am, S. Ozdemir, P. Nair, and D. Muthuavinashiappan, and H.O. Sanli, "Energy-Efficient and secure pattern based data aggregation for wireless sensor networks", Special Issue of Computer Communications on Sensor Networks, pp. 446-455, Feb. 2006.

[8] W. Du and J. Deng and Y. S. Han and P. K. Varshney, "A Witness-Based Approach for Data Fusion Assurance in Wireless Sensor Networks", in Proc. GLOBECOM'03, pp. 1435-9, 2003.

[9] K. Wu, D. Dreef, B. Sun, and Y. Xiao, "Secure data aggregation without persistent cryptographic operations in wireless sensor networks", Ad Hoc Networks, vol. 5, no.1, pp. 100-111, 2007.

[10] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks", Proc. of ACM MOBIHOC'06, May 2006.
[11] S. Ozdemir, "Secure and Reliable Data Aggregation for Wireless Sensor Networks", LNCS 4836, H. Ichikawa et al. (Eds.), pp. 102-109, 2007.
[12] D. Westhoff, J. Girao, M. Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution and Routing Adaptation", IEEE Transactions on Mobile Computing, Vol. 5, No. 10, pp. 1417-1431, October 2006.
[13] S. Ozdemir, "Concealed Data Aggregation in Heterogeneous Sensor Networks using Privacy Homomorphism", Proc. of ICPS'07 : IEEE International Conference on Pervasive Services, pp. 165-168, Istanbul, Turkey, 2007.
[14] C. Castelluccia, E. Mykletun, G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks", Proc. of Conference on Mobile and Ubiquitous Systems: Networking and Services, pp.109-117, 2005.
[15] D. Wagner, "Cryptanalysis of an Algebraic Privacy Homomorphism", in Proc. Sixth Information Security Conf. (ISC03), Oct. 2003.
[16] R.L. Rivest, L. Adleman, and M.L. Dertouzos, "On Data Banks and Privacy Homomorphisms", Foundations of Secure Computation, pp. 169- 179, 1978.
[17] Crossbow Technologies Inc., http://www.xbow.com.
[18] J. Domingo-Ferrer, "A Provably Secure Additive and Multiplicative Privacy Homomorphism", in Proc. Information Security Conf., pp. 471- 483, Oct. 2002.
[19] N. Koblitz, A. Menezes and S. Vanstone, "The State of Elliptic Curve Cryptography", Journal of Designs, Codes, and Cryptography, vol. 19, pp. 173-193, March 2000.
[20] D. Boneh, Eu-Jin God, and K. Nissim, "Evaluating 2-DNF Formulas on Cipertexts", Proc. Theory of Cryptography Conference, LNCS vol. 3374, pp. 325-321, Jan 2005.
[21] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, "A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge", IEEE Transactions on Dependable and Secure Computing, vol.03, no.1, pp. 62-77, January-March, 2006.
[22] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, "Handbook of Applied Cryptography", p. 128, CRC Press, 1997.
[23] S. Peter, D. Westhoff, and C. Castelluccia, "A Survey on the Encryption of Convergecast-Traffic with In-Network Processing", IEEE Transactions on Dependable and Secure Computing, vol. 99, no. 2.